Zhang, Jinglan and Pham, Binh L. and Chen, Yi-Ping Pheobe (2001) A fuzzy genetic algorithm for creative shape design. In *Proceedings Tools and Methods of Competitive Engineering (TMCE2002)*, pages pp. 315-329, Wuhan, China.

A FUZZY GENETIC ALGORITHM FOR CREATIVE SHAPE DESIGN

Jinglan Zhang
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434 Brisbane QLD 4001 Australia
jinglan.zhang@qut.edu.au

Binh Pham and Phoebe Chen
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434 Brisbane QLD 4001 Australia
(b.pham, p.chen)@qut.edu.au

ABSTRACT

Design is a multi-criteria decision-making process under multiple constraints. In the conceptual design stage, design is intrinsically imprecise because of designers' vague thinking and incomplete initial information. When exploring possible design candidates, designers are generally more interested in sets of the most promising solutions rather than the best single solution. Therefore, in contrast to conventional optimisation approaches that aim to find exact optimal points, we aim to find optimal set of alternatives with variable satisfaction degrees. A fuzzy-set-based approach for representation and optimisation of design objects is particularly suitable for solving this problem. The concept of a fuzzy shape is defined as a family of shapes with similar properties where a fuzzy solid shape is represented by a set of parameters that have fuzzy set values. Evolutionary computation is used to obtain fuzzy solutions to the fuzzy shape optimisation problem since it is the most powerful tool for supporting creative design through multiple objectives, multi-dimensional searching. The representation of fuzzy sets, its initialisation, crossover, mutation, and validation, the ranking approach for fuzzy shapes, and the propagation method of fuzzy information are discussed. A case study for illustrating this fuzzy design approach is provided.

KEYWORDS

Fuzzy shape, fuzzy shape evolution, fuzzy genetic algorithm, imprecise design.

1. INTRODUCTION

Design, especially at conceptual stage, is intrinsically imprecise (Antonsson and Otto 1996), where the imprecision comes from both designers' thinking and practical problems. At this stage, designers have only vague ideas that need to be gradually refined. In addition, it is often not possible to acquire complete information at the beginning of design and decisions must be made on the basis of rough experience. A tool that supports conceptual design should capture such imprecise features.

Design can be considered as an optimisation process that aims to find the best solutions to fulfil certain constraints (Scott 1999). When exploring possible design candidates, designers are generally more interested in sets of the most promising solutions than in the best single solution. A set-based solution is therefore more natural and robust than a crisp solution, both in single-person design and cooperative design. Hence, it is desirable for a design solution to contain set-based information.

The fuzzy set approach that was introduced by Zadeh (Zadeh 1965) is particularly suitable for handling imprecise information by providing a set of solutions with different preference degrees. We therefore choose this approach to address the imprecise and set-based design problem. Another approach is the interval-based design (Huang et al. 1998) where design variables are represented by interval values. Since fuzzy sets are generalizations of intervals and crisp values (Petry and Bosc 1996), fuzzy-set-based design may be considered as a generalization of interval-based design and conventional crisp-value-based design.

A fuzzy set of design alternatives is not an arbitrary group of elements because different members in a fuzzy set have similarities to some extent and the differences between members are measurable. A fuzzy set of design alternatives can be considered as a single entity described by their similarities. For example, the term *very round shape* potentially refers to a set of shapes from very round to exactly round (a sphere). Designs that belong to the same fuzzy set and are close in preference are *nearby designs* (Scott 1999). The fuzzy sets of design need to be *connected* so that nearby designs are grouped into one set. In real applications, it is possible to have two or more disjoint sets of desirable designs, but these sets are often considered separately (Berkan and Trubatch 1997).

The applications of fuzzy sets in geometric modelling area are relatively new (Buckley and Eslami 1997a; Buckley and Eslami 1997b; Pham and Zhang 2000; Rosenfeld 1998). We introduced the concept of a fuzzy shape as a set of shapes that looks similar yet slightly different. More information on fuzzy shape can be found in (Pham and Zhang 2000; Zhang et al. 2000a; Zhang et al. 2000b). We are interested in fuzzy-set-based design, i.e., considering a whole set of design alternatives, each of which has a corresponding membership degree. The concept of fuzzy-set-based design, in particular, fuzzy shape design, is embedded into every aspect of the preliminary design, from perceptual shape representation, geometric modelling, calculation, optimisation, to storage and display. We have constructed a fuzzy shape specification system to bridge the gap between fuzzy shape perception and exact geometrical modelling (Pham and Zhang 2000), where each shape is described by a set of fuzzy shape descriptors and modelled by a set of parameters. A fuzzy database for fuzzy shape storage and retrieval has also been constructed (Zhang et al. 2000a; Zhang et al. 2000b). Since a fuzzy shape is represented by a fuzzy set, in the evolution of design alternatives, we not only consider the best designs, but also the nearly best designs. So, in contrast to conventional optimisation approaches that aim to find exact optimal points, we aim to find optimal set of alternatives with variable satisfaction degrees. This paper will focus on an automatic optimisation process of fuzzy shapes.

Optimisation methods can generally be classified into two basic classes: calculus-based (or gradient) method which finds the optimum by analysing the derivative equation and search-based optimisation method which locates the optimum by evaluating function values at suitable search points. The calculus-based method can only be applied to continuous, differentiable functions because it requires information on both the function and the gradient, while the search method uses only function evaluation. Since many design problems are not differentiable under practical constraints, they are not solvable using conventional gradient-based optimisation techniques. We therefore choose a search-based algorithm for shape evolution, in particular, we use a guided random search method, the Genetic Algorithms (GAs), to perform the optimisation task. GAs (Chambers 1995) can solve both uni-modal and multi-modal functions since the setting of search points is based on genetic and evolutionary principles. The application of GAs in design is also consistent to the nature of design. The searching for a good design is actually a trial-and-error process but it is not a random search. It is always guided by the objective targets and subjective preferences. This process is similar to the natural evolution where only the fittest survives. The creation of new design objects through combination is also similar to the genetic principle that better parents tend to generate better children. Since the generation and evaluation of all possible design alternatives through combination is often out of the capability of human, employing GAs to perform shape evolution can simulate the design process.

Many papers have discussed fuzzy optimisation problems using fuzzy coefficients, fuzzy constraints, or fuzzy objective functions. However, almost all existing fuzzy optimisation methods use crisp solutions (Chen and Hwang 1992; Lai and Hwang 1996). Only few attempts considered fuzzy solutions (Antonsson and Otto 1996; Buckley and Feuring 2000; Buckley and Hayashi 1994; Sebastian and Schleiffer 2000). Buckley et al. (Buckley and Hayashi 1994) employed a fully fuzzified genetic algorithm to perform operational research. Their approach uses arbitrary shape of fuzzy sets to get approximate solutions to a fuzzy optimisation problem. It is simple but the resultant fuzzy set is not reasonable because it is not convex. Buckley et al. (Buckley and Feuring 2000) implemented an evolutionary algorithm to solve linear programming problem where only triangular shapes of fuzzy sets are used and the result is good. In both approaches, the traditional fuzzy arithmetic based on the extension principle and convolution is used to propagate fuzzy information. This approach utilises natural set-based information and is suitable for general fuzzy optimisation problems. However, it needs large computation expense and will be impractical in large dimension optimisation problems. Antonsson (Antonsson and Otto 1996) performed a calculus-based optimisation using *one-at-a-time* search which explores all directions one by one, while used preference specifications as fuzzy constraints and Level Interval Algorithm for fuzzy information mapping. This approach relies on gradient information and is suitable only for uni-modal problems that have only one optimum within certain range. Scott (Scott 1999) followed Antonsson's direction but used the *pattern search* to replace the one-at-a-time search in order to avoid the calculation of gradient information. Both Antonsson and Scott's approaches relies on the fuzzy evaluation of crisp points to get the best points with highest preference degree. Sebastian and Schleiffer (Sebastian and Schleiffer 2000) used a crisp genetic algorithm to search the most promising solutions and a fuzzy clustering algorithm to group nearby crisp solutions into fuzzy solutions. This approach can solve multi-modal problems but it has difficulties in defining appropriate number of groups.

This paper aims to investigate techniques for fuzzy shape optimisation under fuzzy constraints using GAs. In particular, we concentrate on the fuzzy representation, fuzzy genetic operations, fuzzy evaluations and orderings of fuzzy shape. We first apply the concept of a fuzzy solution to the fuzzy shape optimisation problem using fuzzy evolutionary computation approach. Section 2 describes the geometric representation of fuzzy shapes. Section 3 discusses general issues considered in shape evolution, while section 4 presents the Fuzzy Genetic Algorithms (FGAs). Section 5 then provides a case study where the fuzzy genetic algorithms are applied to perform an office chair design. Finally, the conclusions and future work are discussed in section 6.

2. FUZZY SHAPE REPRESENTATION

There are many commercial CAD systems used for solid modelling in design, almost all of which are based on exact, detailed representations of objects. Designers must have a clear idea of what each component looks like before specifying objects in computers. However, as design in the conceptual stage is vague and tentative, it is very difficult for the designer to give out an exact description. Hence a simple and flexible modelling method is required. We choose to use the superquadric 3D model proposed by Barr (Barr 1981) to be the basic geometric model because it is simple and easy to control, yet has relatively wide geometric coverage. Therefore, it is suitable for intuitive and interactive shape design at the initial stage.

Superquadrics may be expressed by an implicit equation:

$$\left(\left| \frac{x}{a_1} \right|^{2/\varepsilon_2} + \left| \frac{y}{a_2} \right|^{2/\varepsilon_2} \right)^{\varepsilon_2/\varepsilon_1} + \left| \frac{z}{a_3} \right|^{2/\varepsilon_1} = 1$$

(1)

where, ε_1, ε_2 are shape parameters which control the shape roundness and squareness along north-south direction and west-east direction respectively; a_1, a_2, and a_3 are scalar parameters which represent the length of a shape along x, y, and z axis respectively. Since the shape description of an object does not relate to its size, we let $a_1 = 1$ and a_2, a_3 represent the ratio of a_2/a_1 and a_3/a_1 respectively. The parameters a_2 and a_3 control the relative dimension of the two cross sections of a superquadric shape.

The primary superquadrics cannot create shapes such as a cone, a bent bar or a twisted cube. Hence, several deformation parameters, k_x, k_y, k_v, and t were introduced by Barr (Barr 1984) to control their deformation such as tapering, bending and twisting. Therefore, a shape can be represented by eight shape parameters

$$\{\varepsilon_1, \varepsilon_2, a_2, a_3, k_x, k_y, k_v, t\}$$

where, k_x and k_y control the tapering property; k_v controls the bending property; and t controls the twisting property. This representation is called deformable superquadrics. Further information about deformable superquadrics can be found in (Barr 1981; Barr 1984; Barr 1992; Pham and Zhang 2000). We use deformable superquadrics and still call them superquadrics for simplicity.

Superquadrics provide an algebraic expression of 3D shapes and their specification involves only a few parameters. They are relatively simple and easy to handle, however the process to get a required shape by changing parameters directly is still tedious because the relations between parameters and shapes are not linear. Therefore, we propose to use a set of shape descriptors to specify shape and map them to shape parameters through a fuzzy shape specification system. More details on the shape specification system may be found in one of our previous papers (Pham and Zhang 2000).

Given a set of shape descriptors, such as *very round* and *somewhat square,* a set of shape parameters which represent shapes that correspond to the descriptive terms is obtained from the fuzzy shape specification system. Each shape parameter has a set of values with variable membership degrees. Each specification implies a set of shapes with certain similarities. We call this set of shapes a fuzzy shape.

To sum up, a fuzzy shape is a set of shapes formed by a set of shape parameters with fuzzy set as values. In our system, a fuzzy shape represents a set of crisp superquadric shapes with variable membership degrees. Figure 1 shows several crisp shape elements of a *very bent* fuzzy shape.

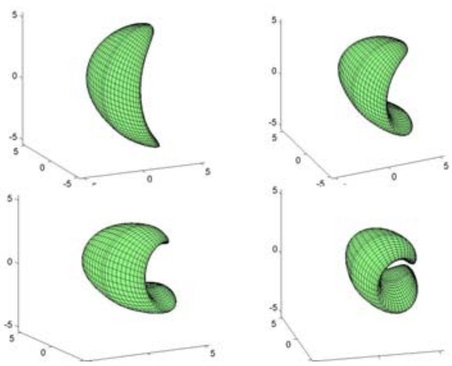


Figure 1. Typical shape elements in a *very bent* fuzzy shape.

3. FUZZY SHAPE EVOLUTION

In the context of solid modelling, design can be classified into structure optimisation for configuration design and parameter optimisation for shape element design. We focus on shape element design that aims to optimise some objectives under certain constraints.

A good design is usually obtained by iterative refinements through tedious trial-and-error with the help of a computer simulation package. GAs potentially allow us to automate this adjustment process through multiple objective, multi-dimensional searching. Since fuzziness exists in the conceptual design stage, it is desirable to retain the fuzzy nature of shapes after performing the shape optimisation. Since a fuzzy shape is represented as a string of fuzzy sets consisting of possible values of shape parameters with different certainty degrees, the optimisation of a fuzzy geometric shape is mapped to the searching of optimal fuzzy values of these shape parameters. The fuzzy shape optimisation is performed through range-by-range searching in the design space using a fuzzy genetic algorithm. Changing the values of shape parameters in superquadric models affects the form itself, so we can explore all shape elements that are within the representation capability of superquadrics. We call this process “ fuzzy shape evolution”.

3.1. Multiple criteria for shape evolution

Design is not only a problem solving activity but also an art creation process. In the competitive design market, a good design with high quality needs not only to satisfy many hard constraints (or objectives) such as functional efficiency, ergonomic comfort, manufacturing easiness, and cost effectiveness, but to meet some soft constraints such as attractive appearance and psychic acceptability. Since a design object is usually represented by a set of design variables, a design problem is typically a constrained multidimensional optimisation problem with multiple objectives.

Since the aesthetic or psychic constraints are highly subjective and fuzzy, it is natural to represent these constraints using a fuzzy set. The ergonomic constraints are derived from statistical data, usually represented by interval values. Since a fuzzy set may be viewed as a generalization of an interval set, we can transform these interval sets to fuzzy sets. As fuzzy-set-based design solutions are more natural and tolerant than conventional crisp solutions, the metric variables such as volume and constraints exerted on these variables are represented as fuzzy sets as well.

3.2. Formulation of objective function

Shape modelling is a highly constrained optimisation problem. One approach to solve this kind of problems is to use a function that contains penalty terms. For example, we can employ the following domain constraint penalty to ensure the shape parameters to remain in allowable region:

$$p_j = \sum_i error(V_i) = \sum_i (1 - \mu(V_i))$$

where $\mu(V_i)$ is the membership degree to which the jth parameter value belongs to a predefined preference range i. The fitness function with penalties can be defined as:

$$f_{fitness} = fitness - \sum_{j=1}^{n} w_j * p_j \ ,$$

where $f_{fitness}$ is the penalised fitness value, *fitness* is the objective function without penalty and is application dependent, p_j is a penalty and w_j is the weight of importance of this penalty.

Constraints that are not directly computable, such as aesthetic requirements, can be mapped to objective constraints on geometric form based on predefined rules. These objective constraints are then mapped to constraints on design variables through a fuzzy inference system. The aggregation of the constraints on the same variable can be performed by the standard intersection operator, *min*, which is proposed by Zadeh (Zadeh 1965; Zadeh 1999).

In GAs, only one objective function is used to obtain a total ordering for all individuals to do automatic evolution. Hence, we need to combine multiple objectives into one single objective. There are many ways to perform this aggregation depending on the nature of the objectives. Some objectives are not compensating, for example, in engineering design, the minimum structural requirements and material requirements must be satisfied at the same time. Generally the *min* aggregation approach can be used to measure the overall satisfaction degree for these cases. If the objectives are compensating, the weighted multiplication or addition can be employed to calculate the overall fitness. If some of the objectives are compensating and others are not, a combination of the above aggregation approaches can be used to derive a complex fitness function. Detailed discussion on the compensating or non-compensating aggregation approaches can be found in (Scott 1999). A thorough overview and critical analysis of the aggregation approaches can be found in (Lai and Hwang 1996). We choose the Fuzzy Simple Additive Weighting Methods (Lai and Hwang 1996) to aggregate multiple objectives into one objective for the shape evolution. The basic assumption for this approach is that different objectives can compensate to each other. This is true for evaluation of a design product based on non-technical hard requirements such as those in industrial design. For example, if the cost is not the minimum, but the aesthetic value is very high, a product may also be considered better than a product that has the minimum cost but is less beautiful. The settings of the weights are also very subjective in this case.

In the fuzzy simple additive weighting approach, the performance function of a design alternative can be formulated as follows:

$$U = \sum_{j=1}^{n} w_j r_j \Big/ \sum_{j=1}^{n} w_j$$

where U is the overall fitness value of the alternative, r_j is its fitness value under jth objective, w_j is the weight of the jth objective, and n is the number of objectives. The alternatives with the highest overall fitness value are the best.

In shape evolution, the weights can be crisp numbers or fuzzy linguistic values depending on the source of data. We choose to use crisp numbers for simplicity. Since shape parameters have fuzzy values, all component objective values and the overall objective value are all fuzzy sets. The evaluation of the objective function is performed by fuzzy arithmetic based on alpha-cuts of fuzzy sets, which are subsets of all elements with the membership grades higher than alpha. In particular, we use the Level Interval Algorithm (LIA) (Antonsson and Otto 1996) to perform fuzzy computation. The LIA uses the alpha-cut end points of all variables to construct the alpha-cut bounds of the resultant fuzzy set. Once all the performance fuzzy sets for all possible alternatives are obtained, the alternatives can be ranked according to their performance based on a fuzzy ranking approach. Details on alpha-cuts, LIA and fuzzy ranking approaches will be discussed in a later section.

In summary, the multiple objectives optimisation problem is transformed into a single objective optimisation problem through fuzzy simple additive weighting. When the design constraints on one variable are obtained from a fuzzy inference system, the *min* approach is used to aggregate multiple constraints into one constraint. Then, all constraints on all variables are incorporated into the corresponding objective functions through penalty functions.

3.3. Information mapping between different design spaces

A knowledge-based system is useful for supporting design since the design process involves a substantial amount of existing knowledge. Some knowledge such as structural dimension is shallow knowledge while other knowledge such as aesthetic evaluation or other functional properties is deep knowledge. Some knowledge is actual values which designers can manipulate and others is expected values from customers or practical limitations. Some knowledge is concerned with independent parameters while others come from derived properties. Hence, it is necessary to divide the design knowledge into different levels in order to manage it efficiently. There are many possible schemes for this classification (Gero 1990) and we discuss the two most popular schemes here.

The Method of Imprecision (MoI) proposed by Antonsson (Antonsson and Otto 1996) is a formal method for modelling the imprecision information in preliminary engineering design. In MoI, an engineering design problem is modeled by two spaces: *design variable space* and *performance variable spaces*. Design variables (or input parameters) are independent parameters whose values are determined during the design process and performance variables (or output parameters) are parameters whose values are dependent on design variables or other performance variables. The preferred values of these variables are represented by fuzzy sets. The LIA is used to map the fuzzy information from design variable space to performance variable space. We use the fuzzy modelling of design variables and the LIA because they are suitable for modelling and propagating imprecise information in design. However, since the design variable space and performance variable space represent design in two basic levels, it cannot conveniently represent higher level knowledge, such as aesthetic knowledge or general functional requirements in design.

Another design modelling scheme, the Function-Behaviour-Structure (FBS) design framework, is proposed by Gero (Gero 1990). The FBS scheme divides design into three state spaces: the function space, behaviour space, and the structure space. The structure space describes the geometric elements and configuration of an artefact; the behaviour space is the derived properties of an artefact; and the function space defines the purposes of a design object. For example, in an office chair design, the shape parameters within certain domains form the structure space. The geometric appearance such as *squarish* or *very bent* and the volume or surface area belong to the behaviour space. The functional descriptions such as aesthetic properties or ergonomic properties compose the function space. In the design process, the three spaces are related through an input-output relationship or a network, one space can be transformed into another. The design variable space and performance variable space in MoI correspond to the structure space and behaviour space in FBS. Since the aesthetic properties of a product play an important role in a successful design, we aim to incorporate these constraints into the design process. As the FBS structure is more suitable for mapping the aesthetic requirements to geometric descriptions, then to geometric definition variables, or vice versa, we use this scheme to model the design space.

The mapping of non-computational information, such as the mapping from aesthetic requirements to geometric descriptions, is performed by symbol matching. The mapping from geometric descriptions to shape parameter requirements is achieved by a fuzzy inference system where the underlying knowledge is predefined by experts.

The mapping function of imprecise design information from design space to performance space (or behaviour space) is essential in the fuzzy optimisation problem since it determines the relationship between input and output variables. The mapping function can be any computational algorithms, for example, a simple function $a = b+c$ or a complicated finite element analysis program. It can also be a procedure such as a rule-based fuzzy inference system. If the mapping function is a computational algorithm, it is relatively simple and can be performed by the Level Interval Algorithm (LIA) (Antonsson and Otto 1996). We can construct the fuzzy set of the surface area of a shape by finding its minimum and maximum values in each alpha-cut level through the LIA. We use a cellular model, also called Finite Element Model (FEM), to calculate the surface area for a crisp superquadric shape. This approach is suitable for the calculation of many physical properties for both elemental shapes and complex shapes that are composed of many shape elements.

In the next section, we discuss the FGA from the point of view of shape evolution. More information on the FGA can be found in (Zhang et al. 2001).

4. FUZZY GENETIC ALGORITHM FOR SHAPE EVOLUTION

The combination of Genetic Algorithms (GAs) and Fuzzy Systems has been explored for many years (Cordon et al. 1997) but most cases deal with simply coupled systems. They either use GAs to design fuzzy systems such as optimizing membership functions or extracting fuzzy inference rules, or use a fuzzy system to tune GAs parameters such as population size, crossover rate and mutation rate. Systems that integrate these two concepts are rare. Our fuzzy genetic algorithm is an integrated system because the representation, crossover, mutation and selection are all based on the concept of fuzzy set.

We call this algorithm a Fuzzy Genetic Algorithm (FGA) because each individual is composed of a set of fuzzy subsets of real numbers. It is essentially a fuzzy real coded genetic algorithm. The differences between this algorithm and a conventional genetic algorithm are: the representation of individuals; the crossover and mutation operations; the validation of newly generated fuzzy sets; the evaluation of the objective function using fuzzy computation through LIA; the calculation of the satisfaction degree to fuzzy constraints; and the ordering approach based on fuzzy sets representation.

4.1. Representation of fuzzy shapes in the FGA

Since a fuzzy shape is represented by a set of parameters that have real values and all of them are represented by fuzzy sets, the fuzzy sets used in this paper are fuzzy subsets of real numbers. The representation of fuzzy shapes in the FGA consists of two parts: the representation of fuzzy sets and the encoding of fuzzy shapes in each individual.

Representation strategies of fuzzy sets

There are three representation schemes of fuzzy sets used in the FGA: piecewise linear representation, equal interval distribution representation, and family of α-cuts representation. The piecewise linear representation uses a few data points to represent piecewise linear functions such as triangles or trapezoids. In the shape evolution process, this representation is used to represent the designer's preference since this is the most simple and intuitive way to specify a fuzzy set. The equal interval distribution approach represents a continuous fuzzy set using equally distributed discrete points. When the universe of discourse and the number of intervals of a fuzzy set are fixed, the location of each point is fixed. Hence, the location information need not be represented and a fuzzy set can be represented by a vector of membership grades. This representation is suitable for representing an arbitrary shape of fuzzy set because only the membership grades are represented. It is used for randomly generating new fuzzy sets and encoding fuzzy sets into the FGA. The α-cut of a fuzzy set is the subset in which the membership grades of all elements are equal to or greater than the value α. The term α-cuts and alpha-cuts are used interchangeably. The α-cuts approach represents fuzzy sets using α-cut endpoints at a set of predefined α-levels. Since the α-cut values are predefined and fixed, they need not be represented for every fuzzy set. Hence, a fuzzy set can be represented by a set of pairs of α-cut end points. In the FGA, the family of α-cuts is used for propagating fuzzy information using the LIA. It is also used to smoothen an arbitrary shape of fuzzy set to make it alpha-convex which means the result of any strong α-cuts of the fuzzy set are connected sets.

These three discrete representations have a common characteristic that they represent fuzzy sets using discrete points. Since they are different representation schemes for the same fuzzy sets, they can be transformed from one form to another.

Encoding of fuzzy shapes in the FGA

Since a fuzzy shape is represented by a set of parameters that have fuzzy set values, each individual of the genetic algorithm is composed of a set of fuzzy sets. Each fuzzy set is represented by equal interval distribution in order to explore different shape of fuzzy sets. Each gene is a fuzzy set represented by a vector of membership grades that are real numbers between 0 and 1. For example, if two fuzzy sets *A* and *B*, each of them is represented by an equal interval possibility distribution that has five elements, A = {**0.1, 0.4, 1.0, 0.3, 0.0**} and B = {0.2, 0.9, 1.0, 0.6, 0.0}, then the chromosome composed of these two fuzzy sets is represented by I = {**0.1, 0.4, 1.0, 0.3, 0.0**, 0.2, 0.9, 1.0, 0.6, 0.0}. The major advantage of this representation scheme is that it can represent any shape of fuzzy sets.

4.2. Population initialisation

As GAs are essentially random searching processes, except for the definition of the problem, they do not need much knowledge from human. However, the incorporation of human knowledge or the incorporation of existing rough optimum results from other optimisation approaches, can definitely improve and speed up the evolution process in GAs. Thus, existing knowledge is often applied to the initialisation step to add more deterministic knowledge into this random optimisation process.

Initialisation approaches can be classified into three categories: random initialisation, deterministic initialisation and a combination of random and deterministic initialisation. In random initialisation, all individuals are sampled randomly within the valid domain. This is the most popular initialisation approach which utilizes the power of automatic search to the extreme extent but the searching process may be slow, especially when the search space is very large. The deterministic (or heuristic) initialisation approach assigns all individuals to preferred solutions, and usually all good schemata have at least one representative in the initial population. This approach can incorporate expert knowledge or good results from other optimisation method into GAs. It can speed up the searching process but easily leads to premature convergence. In the combined initialisation, one part of the individuals are generated according to existing preferences while the other part of individuals are generated randomly.

Since we need to incorporate designers' experience into the shape evolution process, but designers usually cannot specify all individuals needed in the initial population, we use the combination of random initialisation and deterministic initialisation. Although the initial population is not fully deterministic, partial incorporation of existing knowledge can also improve the evolution process to some extent. We assign very high fitness value to the preferred solutions and always pass best individuals in the current population to the next population in order to retain and propagate a designer's initial preference.

4.3. Crossover and mutation of fuzzy shapes

Since a fuzzy shape is represented by a string of fuzzy sets, the crossover and mutation of fuzzy shapes are based on fuzzy sets. We use the discrete equal interval distribution representation of fuzzy sets for all parameters of fuzzy shapes in GA encoding. Since the location information is not included in the chromosome, all data in the chromosome are membership degree values which are between [0, 1], hence these values can be shuffled arbitrarily as that in binary genetic algorithms, they can also be changed gradually as the real number creep by adding a certain value to the original one (Herrera et al. 1998). Therefore the crossover and mutation in fuzzy genetic algorithms share both the features of binary-coded genetic algorithms and real-coded genetic algorithms. Existing crossover and mutation approaches can be applied to fuzzy genetic algorithms but the encoding and decoding approaches need to be modified to correctly interpret the meaning of a fuzzy set. More detailed discussion on fuzzy crossover and mutation can be found in (Zhang et al. 2001). Here, we focus on the physical meanings of crossover and mutation in shape evolution.

Crossover of superquadric shapes

Since the evolution of fuzzy shapes is based on fuzzy sets and a fuzzy shape is a set of shapes that need to be displayed on multiple views, we use two crisp shapes to show the meanings of crossover process during shape evolution. The crossover of two shapes exchanges the values of definition parameters to generate new shapes. It corresponds to the recombination process of existing design solutions to create new design. Figure 2 shows the crossover result from the top two shapes into the bottom two shapes when they exchange their tapering, bending, and twisting shape parameters. The physical meaning of fuzzy shape crossover is the same as the crisp shape crossover. The only difference is that the crisp values are replaced by fuzzy sets.

Mutation of superquadric shape

The mutation of a shape can change the shape appearance gradually or steeply depending on the mutation step size. Figure 3 shows the three mutation results from the top-left superquadric shape when it changes its tapering and bending shape parameters with different step sizes.

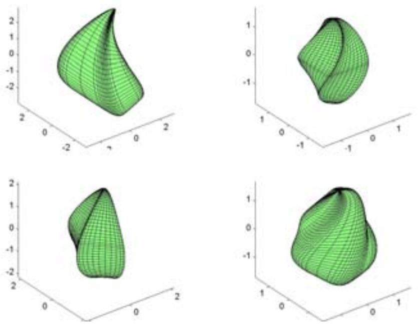


Figure 2. Shape crossover.

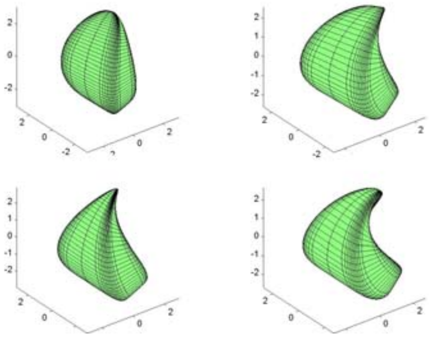


Figure 3. Shape mutation

4.4. Fuzzy shape evaluation

In genetic algorithms, the evaluation of individuals is usually performed by the evaluation of a fitness function which is defined within certain domain. Designers can also perform the evaluation based on experience and subjective preference. Human evaluation is necessary when the fitness function is very hard to define and relies on interactive techniques. Such method for human evaluation should be as simple as possible because it invokes heavy burden. We use the fitness function as the main evaluation approach because a superquadric shape can be mapped to a set of parameters and measured by a function. However, we also allow designers to evaluate the designs and control the evolution process through a Graphical User Interface. In each generation, shapes are automatically generated and evaluated, the best shape and its corresponding fitness value is displayed on the screen, so designers can examine and feel it. If a user finds good individuals that have no higher fitness value or the evolving process is not good enough, s/he can stop the automatic evaluation process, modify the fitness function, and start the searching process again.

In the fuzzy genetic algorithm, the automatic evaluation of the fuzzy fitness function plays the main role for shape evaluation. The evaluation of a shape consists of three parts: the evaluation of a set of objectives and the evaluation of a set of constraints as well as the aggregation of these evaluation results to form an overall fitness value.

The satisfaction of a fuzzy datum to a fuzzy constraint is measured by the *necessity degree* in possibility theory that is the degree to which a datum necessarily satisfies a constraint. When the necessity degree is 1, we can ensure that all elements in the fuzzy set of the datum are within the fuzzy set of the constraint with the satisfaction degree 1. More information on the calculation of necessity degree can be found in (Zhang et al. 2001).

The evaluation of a fuzzy function which has fuzzy variable values is performed by the Level Interval Algorithm (LIA) (Antonsson and Otto 1996). LIA uses a predefined number of alpha-cuts to compute the fuzzy function value based on fuzzy variables. It is based on the idea that the function surface is determined by the endpoints of alpha-cut in each dimension, where the final function value is obtained by searching the super cube which is composed of all the endpoints of alpha-cuts. This idea is not correct when the surface function has local extrema between the alpha-cut endpoints of one or more direction. This problem is solved by finding the local extrema through a crisp optimisation method and then consider the alpha-cut endpoints. The final function range values are the minimum/maximum of the local extrema and the endpoints. The advantage of LIA over the simple alpha-cut approach in (Kaufmann and Gupta 1985) is that it releases the constraint of convexity of the fuzzy function by considering the local optima and the function values of the endpoints of the alpha-cut. More information on LIA can be found in (Antonsson and Otto 1996).

The LIA is suitable only for normal, monotonic, and alpha-convex membership functions. However, in the fuzzy genetic algorithm, after the genetic operation (crossover and mutation), the newly generated membership functions have arbitrary shapes, so we need to perform *validation checking* to ensure the generated fuzzy sets be valid. After the validation checking we can ensure that every fuzzy set is normal, monotonic, and alpha-convex. More details on this validation checking approach can be found in (Zhang et al. 2001).

During shape evolution, the LIA is used for calculating the geometric properties of fuzzy shapes such as volume or surface area. Since a fuzzy shape is a set of shapes with different membership grades, the geometric properties are also a set of values with variable certainty levels. For each alpha-level, we find the minimum and maximum values of a geometric property and forms the two end points of that property in that alpha-level. We traverse each alpha-level and the fuzzy set of the corresponding geometric property can be obtained.

In section 3.2, we have discussed that constrained optimisation problems can be converted into unconstrained problems using penalty terms. Multiple penalties can also be aggregated together and integrated into the fitness function by simple additive weighting approach that has been discussed before.

4.5. Ranking and selection of fuzzy shapes

The ranking of design alternatives are based on the ranking of their fitness values. The fuzzy shape optimisation problem is to find a set of fuzzy values to maximize or minimize a fuzzy function value. Since the value of a fuzzy function is a fuzzy set, we cannot optimise it directly because fuzzy sets have only partial order. This is a major difference between fuzzy data that are partially ordered and crisp data that are totally ordered. Hence, to get a total ordering of fuzzy sets, a scale measure must be defined according to some criteria.

The criteria used for choosing a ranking method in this fuzzy genetic algorithm for fuzzy shape evolution are as follows:

- The ranking measure should be global, that is, all possible alternatives can be ranked according to the same measure and based on the global scale of an alternative, rather than the local preference such as pair-wise comparison.
- To automate the shape evolution process, the ranking process should be based on a unique ranking score (a crisp value). Those ranking methods that use multiple values for ranking need to be converted to a unique score, usually by simple additive weighting or a combination function defined by the decision maker.
- The ranking method should use as much available information as possible, that is, fuzzy information obtained in the optimisation process should be retained until the last stage for comparison. Earlier defuzzification may lead to the lost of useful information.
- The ranking approach should be suitable for any shape of membership functions since the membership functions of shape parameters can have any shapes that are monotonic, alpha-convex and normal.
- It should be easy for us to incorporate this ranking measure into a genetic algorithm, that is, the calculation of the scalable measure should be as quick as possible because GAs rely on extensive computation.

Since the centroid of gravity of a fuzzy set proposed by Yager (Chen and Hwang 1992) is the most simple and popular measure to defuzzify a fuzzy set, we use this measure to perform quick test of the genetic algorithm. As the centroid of gravity uses only one point of a fuzzy set to measure the whole set, sometimes it gives irrational results. To make the comparison of two fuzzy sets more accurate, we combine both the spread of a fuzzy set and its centroid to define the measure for ordering two fuzzy sets using the simple additive weighting approach. The underlying assumption of this ordering approach is that the fuzzy set with a higher centroid value and within the appropriate fuzziness level is better. More discussions on the measures for ordering fuzzy sets can be found in (Chen and Hwang 1992).

Once the fuzzy ranking approach is defined, the selection of individuals from current population and the reproduction of new population is the same as that in a conventional genetic algorithm. For simplicity, we choose the proportional selection scheme which chooses parent individuals with the probability proportional to its fitness value to perform cross over and mutation. The commonly used Elitism reproduction scheme which copies the best members of a generation to the next generation is used to produce the new population.

4.6. Control of shape evolution process

There are three ways to stop the evolution process. Firstly, a crisp maximum number of generations is predefined and used to control the evolution process. Secondly, corresponding to the use of average fitness value to control the evolution process in conventional genetic algorithms, we also use an average fitness value as another measure to stop the evolution process. The application of artificial intelligence techniques in design is not aimed to replace people, but mainly to provide support and empower people. Hence we develop a Graphical User Interface to allow designers to interact with the program and monitor the evolution process. If it is necessary, designers can terminate the evolution process, modify the fitness function, adjust fuzziness degree, relax or exert some constraints or add more heuristics to the initialisation process. This human-control mechanism forms the third method to stop the fuzzy genetic algorithm. This approach is complementary to the first two automatic stopping criteria and should not be relied on heavily because human evaluation is time and energy consuming.

In summary, a fuzzy shape is represented by a set of shape parameters which are fuzzy sets denoted by equal interval possibility distributions. All these fuzzy sets are concatenated together to form an individual in the fuzzy coded genetic algorithm. The searching process starts from some initially specified fuzzy design alternatives and some random samples from the valid design space. Then it exploits the whole design space through guided searching by crossover and mutation operations. The shapes of membership functions are arbitrarily generated but validated by a checking process to fulfil the general meaning of a fuzzy set. Since the variables in the objective function have fuzzy set values, the individuals are evaluated through a fuzzy information propagation algorithm called Level Interval Algorithm. The ordering of individuals is based on a predefined scale that measures the largeness of a fuzzy set and the selection of parents for mating and the reproduction of the next generation are based on this ranking approach.

The next section provides a real application of the fuzzy genetic algorithm to perform the shape evolution that incorporates both subjective and objective criteria. This example shows the potential power of the fuzzy genetic algorithm in product design.

5. CASE STUDY OF FUZZY SHAPE EVOLUTION

The basic mathematical optimisation problem is to minimize or maximize a scalar value of a function that has multiple parameters, $\min(f(x))$ or max $(f(x))$. The minimization problem can be formulated in terms of maximization problem: $\min(f(x)) = -\max(-f(x))$ and vice versa. Hence, in the fuzzy genetic algorithm, without the lost of generality, we use the minimum optimisation problem to illustrate how this algorithm works. Genetic Algorithms are intrinsically suitable for maximization problem because the fitness function usually has the property that the higher its value, the better the solution. We therefore translate the minimization problem into maximization problem within the fuzzy genetic algorithm.

The design of an office chair is used as an example to show the shape evolution process. Since an office chair is an equipment for human use, it needs not only to suit human's dimensions and biomechanical capabilities, but also to satisfy their psychological needs. In addition, the design of an office chair needs to fulfil the economic requirements in order to benefit the developer and many other constraints. As an illustration of how to apply the fuzzy genetic algorithm in design, we do not list all the objectives and constraints met in real design. Instead, we use only several typical objectives and constraints to show the working process. In particular, we use the metric objectives and aesthetic constraints. Since the aesthetic constraints are highly subjective and fuzzy, it is natural to represent these constraints using fuzzy sets. Since the metric properties such as volume or surface area are derived properties of a fuzzy shape that has fuzzy parameter values, they are fuzzy sets as well. Since fuzzy shapes have fuzzy set as parameter values, in contrast to crisp point searching in conventional optimisation approach, the searching process in fuzzy genetic algorithm is based on fuzzy sets and the solution is a set of fuzzy sets. Hence, in the fuzzy optimisation problem, we have fuzzy objectives, fuzzy constraints and fuzzy solutions.

An office chair is usually composed of many parts such as a back, a seat, a back support, armrests and a base. Since we focus on shape element design, we choose the design of the back of a chair as an example. Since the design process is a multi-dimensional searching with multiple objectives, we first need to identify the objectives that will be achieved and the constraints that will be satisfied.

The first objective for the design of the back of an office chair is to minimize its surface area. Suppose p is a vector of real number fuzzy sets representing the values of shape parameters, we have:

Objective 1: *min*(*surfaceArea*(*p*)).

Another objective is related to the property of fuzzy solutions. We intuitively wish that the fuzziness of each fuzzy solution is maintained in a certain level. If it is too narrow (in extreme, a crisp value), we lose the effect for fuzzy optimization. However, if it is too wide (in extreme, the whole valid domain of a variable), it lose its selectivity. Hence we aim to retain the spread of every fuzzy solution in a reasonable range. This forms our second objective:

Objective 2: *max*(*sum*(*fuzzy sets with valid spread).*

According to the experiment results reported in (Jindo et al. 1995), regarding to the shape design of the back of an office chair, the word *stylish* means:

"The front view of the back is *very squarish*, the surface curve of the back is *slightly curved*, and the thickness of the back is *very thin*."

These requirements form the following constraint:

Constraint 1: *stylish* → *squarish, slightly curved,* and *thin.*

We then map these words into shape parameter ranges through the fuzzy inference system and derive a set of fuzzy constraints on shape parameters, each of which is represented by a piece-wise linear membership function:

ε_2 = {1.0/0.0, 0.3/0.25, 0.0/0.35},

k_v = {0.0/0.0, 1/0.02, 0.0/0.2}, and

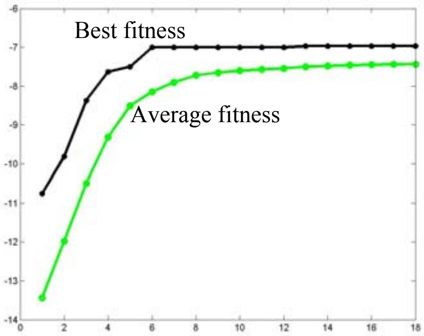
a_3 = {0.0/1, 1.0/10, 1.0/30, 0.0/40}.

The first two optimising targets are objective and we can incorporate them to the fitness function directly using the simple additive weighting approach. Since the aesthetic constraint is subjective and not directly related to the design variables, we first map them to corresponding geometric descriptors by symbol matching. We then map the geometric descriptors to the expected fuzzy set values of design variables through a fuzzy inference system where the knowledge relating the subjective descriptions to the design variable values are predefined. More details on the fuzzy inference system can be found in (Pham and Zhang 2000). Through the fuzzy inference, we obtain the expected fuzzy sets for shape parameters and we use them as fuzzy constraints in the fuzzy genetic algorithms. These fuzzy constraints are then incorporated into the fitness function through penalty functions which have been discussed in section 3. Figure 4(a) shows the evolution process of the FGA, where the top curve is the best average over generations and the bottom one is the average fitness. Figure 4(b) shows the fuzzy solutions of the shape parameters. Since the FGA is quite slow when the searching dimension is high, we optimise four shape parameters and fix the others. Figure 4(c) displays the typical shape elements of the resultant fuzzy shape.

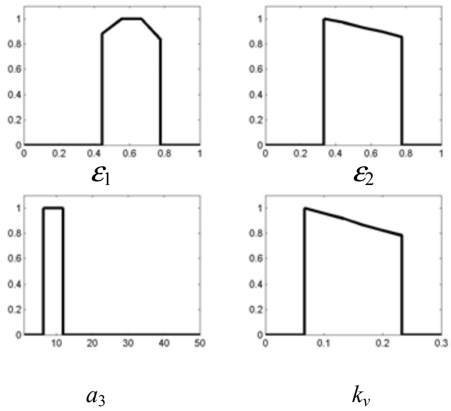
6. CONCLUSIONS AND FUTURE WORK

Effective modelling and propagation of imprecise information is necessary in conceptual shape design that is intrinsically imprecise. The modelling of imprecise information is also motivated in the interest of finding a set of variants that fulfil certain requirements. The fuzzy design approach based on the concept of fuzzy shape can cater for these imprecise features.

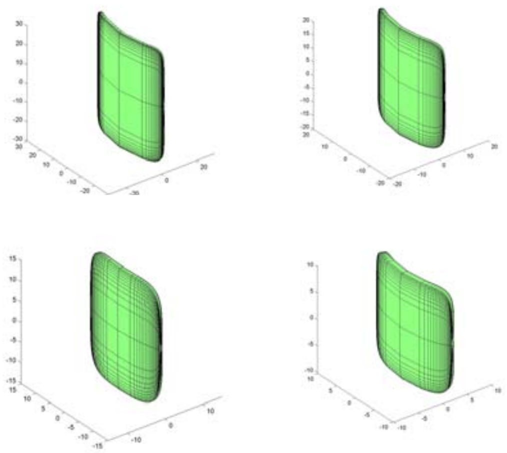
When exploring possible design candidates, designers are generally more interested in sets of the most promising solutions than in the best single solution. This fuzzy genetic algorithm uses fuzzy solutions to a fuzzy optimisation problem through range-by-range searching. It not only finds the best single solution to an optimisation problem, but also plenty of promising alternatives that are nearly best and naturally graded. The experimental results show that this algorithm can produce good approximate solutions. The LIA for fuzzy computation is quicker than the conventional fuzzy arithmetic based on a convolution approach. However, LIA is still time-consuming when the dimension searched is high. We can also conclude that the incorporation of existing experience into the initialisation process of GAs speeds up the optimisation process although in theory, it may cause the so-called “premature convergence”, i.e, only the preferred local optimal points are found and the global optimal solutions left unexplored.



(a) Best and average fitness over generations



(b) Fuzzy set values for shape parameters



(c) Typical shape elements in the *squarish, slightly curved,* and *thin* fuzzy shape

Figure 4. Shape element optimisation results

Future work includes investigating more effective approaches for fuzzy information propagation since current LIA approach is time-consuming when the dimension searched is high. The fuzzy genetic algorithm needs also to be extended to cater for the evolution of complex fuzzy shapes composed of fuzzy shape elements.

REFERENCES

Antonsson, E. K., and Otto, K. N. (1996). "Improving Engineering Design with Fuzzy Sets." Fuzzy Sets in Engineering Design and Configuration, H.-J. Sebastian and E. K. Antonsson, eds., Kluwer, 1-52.

Barr, A. H. (1981). "Superquadrics and Angle-preserving Transformations." *IEEE Computer Graphics & Appl.*, 1, 11-23.

Barr, A. H. (1984). "Global and Local Deformations of Solid Primitives." *Computer Graphics*, 18(3), 21-30.

Barr, A. H. (1992). "Rigid Physically Based Superquadrics." *Graphics Gems III*, 137-148.

Berkan, R. C., and Trubatch, S. L. (1997). *Fuzzy System Design Principles*, IEEE Press, NY.

Buckley, J. J., and Eslami, E. (1997a). "Fuzzy Plane Geometry II:Circles and Polygons." *Fuzzy Sets and Systems*, 87(1), 79-85.

Buckley, J. J., and Eslami, E. (1997b). "Fuzzy Plane Geometry I:Points and Lines." *Fuzzy Sets and Systems*, 86, 179-187.

Buckley, J. J., and Feuring, T. (2000). " Evolutionary Algorithm Solution to Fuzzy Problems: Fuzzy Linear Programming." *Fuzzy Sets and Systems*, 109, 35-53.

Buckley, J. J., and Hayashi, Y. (1994). "Fuzzy Genetic Algorithm and Applications." *Fuzzy Sets and Systems*, 61, 129-136.

Chambers, L. (1995). *Practical handbook of genetic algorithms*, CRC Press, Boca Raton.

Chen, S. J., and Hwang, C. L. (1992). *Fuzzy Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, New York.

Cordon, O., Herrera, F., and Lozano, M. (1997). "A Classified Review on the Combination Fuzzy Logic-Genetic Algorithms Bibliograpgy: 1989-1995." Genetic Algorithms and Fuzzy Logic Systems : Soft Computing Perspectives, E. Sanchez, T. Shibata, and L. A. Zadeh, eds., World Scientific Pub, Singapore ; River Edge, NJ, 209-240.

Gero, J. S. (1990). "Design prototypes: a knowledge representation schema for design." *AI Magazine*, 11(4), 26-36.

Herrera, F., Lozano, M., and Verdegay, J. L. (1998). "Tackling Real-Coded Genetic Algorithms:

Operators and Tools for Behavioural Analysis." *Artificial Intelligence Review*, 12, 265-319.

Huang, W., Liu, J., Zhang, X., and Zhou, J. (1998). "A Qualitative Approach to Solid Modeling for Conceptual Design." *Journal of CAD&CG (China)*, 10(Supplement), 185-.

Jindo, T., Hirasago, K., and Nagamachi, M. (1995). "Development of a Design Support System for Office Chairs Using 3-D Graphics." *International Journal of Industrial Ergonomics*, 15, 49-62.

Kaufmann, A., and Gupta, M. M. (1985). *Introduction to Fuzzy Arithmetic : Theory and Applications*, Van Nostrand Reinhold Co., New York, N.Y.

Lai, Y.-J., and Hwang, C.-L. (1996). *Fuzzy Multiple Objective Decision Making: Methods and Applications*, Springer-Verlag, Berlin Heidelberg.

Petry, F. E., and Bosc , P. (1996). *Fuzzy Databases: Principles and Applications*, Kluwer Academic Publishers, USA.

Pham, B., and Zhang, J. (2000). "A Fuzzy Shape Specification System to Support Design for Aesthetics." Soft Computing in Measurement and Information Acquisition, L. Reznik, ed., Physica-Verlag, Heidelberg, in print.

Rosenfeld, A. (1998). "Fuzzy geometry: An updated overview." *Information Sciences*, 110, 127-133.

Scott, M. J. (1999). "Formalizing Negotiation in Engineering Design," Ph.D. Thesis, California Institute of Technology, Pasadena, California.

Sebastian, H.-J., and Schleiffer, R. (2000). "Using Computational Intelligence in Fuzzy Engineering Design." *Cybernetics and Systems*, 31(5).

Zadeh, L. A. (1965). "Fuzzy Sets." *Information and Control*, 8, 338-353.

Zadeh, L. A. (1999). "Fuzzy Logic=Computing with Words." Studies in Fuzziness and Soft Computing: Computing with Words in Information/Intelligent Systems 1: Foundations, L. A. Zadeh and J. Kacprzyk, eds., Physica-Verlag Heidelberg.

Zhang, J., Pham, B., and Chen, P. (2000a). "A Fuzzy Shape Database for Creative Design: Technical Report." *FIT-TR-2000-02*, Faculty of Information Technology, Queensland University of Technology, Brisbane, Austalia.

Zhang, J., Pham, B., and Chen, P. (2000b). "Indexing and Retrieval of Fuzzy Shapes." Fuzzy Logic: a Paradigm for the New Millennium, V. Dimitrov and V. Korotkich, eds., Physica-Verlag, to appear, Mackay, Australia.

Zhang, J., Pham, B., and Chen, P. (2001), "Fuzzy Genetic Algorithms Based on Level Interval Algorithm." *The 10th IEEE International Conference on Fuzzy Systems*, Melbourne, Australia, accepted.